说点题外话03｜银弹可以杀死狼人，但你怎么知道狼人不是你呢？

2021-07-24 徐昊

《如何落地业务建模》 课程介绍 >

讲述：徐昊
时长 04:35 大小 4.21M

你好，我是徐昊。今天我们再来专门说点题外话。

在软件开发的黑话里，有一颗银色子弹（并不是滚筒洗衣机）可以解决一切问题，而我们一代代软件人，都在苦苦追求它。每当有新技术出现的时候，就会有人问，XXX 是不是银弹啊？比如说啊，云计算是不是银弹，DDD 是不是银弹，RESTful API 是不是银弹，低代码是不是银弹（并不是，这是行业毒瘤）。

然而有意思的是，银弹这个隐喻被引入软件开发领域中的时候，是源自 Fred Brooks 经典的软件工程论文《没有银弹：软件工程的本质性与附属性工作》（*No Silver Bullet Essence and Accidents of Software Engineering*）。

简而言之，Fred Brooks 将软件开发中的工作分为本质性工作（Essential Task）和附属性工作（Accidential Task）。所谓本质工作，就是解决本质性困难的工作。而软件的本质性困难就是：**如何从抽象性问题发展出具体概念上的解决方案**。也就是如何理解我们要解决的问题，并选择恰当的解决方案。

与之相对的则是附属性工作，也就是将寻找到的解决方案，转化为电脑可执行程序的工作。而在这个过程中遇到的困难，就是附属性困难。

Fred Brooks 说呢，如果我们把具体的软件开发看成由本质性工作和附属性工作构成的混合形态，只要附属性工作不是占据全部工作的 9/10，那么就算我们把附属性工作降到 0，也没法获得十倍的效率提升。

也就是说，理解问题并寻找解决方案的本质复杂度难以降低，它一定会占据软件开发中的大部分时间和精力（比如知识消化的“两关联一循环”）。所以哪怕一行代码都不写，我们也没有办法将生产效率提升十倍。

这告诉我们什么呢？当你思考你的职业生涯时，需要考虑一下，你的本质性技能和附属性技能。然后我想你们都懂本质性技能是什么，因为我们已经强调了无数次了，那就是理解问题、定义问题。

我曾经领导过我司一个技术管培生计划，目的是培养十倍效能的 Thoughtworker，也叫小巨人计划。考虑到我司员工略高于行业平均水平的效能，当时就有人心怀疑虑，在这个基础之上，怎么还能再提高十倍呢？

其实答案很简单，任何人只要学会工作中的时间管理和人格分裂性多人运动，也就是任务分解和测试驱动开发，都能获得三倍左右的效率提升。而想要再进一步，就需要有更好的定义问题的方法，从本质性任务上寻找突破。这就需要我们能更好地理解业务，并且能够快速地学习领域知识。

前者可以通过学习财务、法务知识入手，并通过四色建模以及后续 8X Flow 在工作中的强化，后者则需要通过刻意练习，训练对应的胜任力。这其实不难，但难的是转变思路和行为。正所谓易学难精，这是需要你自己多下功夫的，我在这里就不多展开了。

当然，对于 Fred Brooks 的论断也有一种逆向推理，也就是能不能将附属性工作的比例提高，高到超过 9/10 呢？**然后发现方法竟然惊人地简单：不定义问题、随意归因和迷信复用。**

不定义问题就是不问要解决什么问题，按着套路推代码就行了。反正最后落到代码上也不过就是 CRUD，那管它到底要 CRUD 啥呢。于是乎，别说 9/10 了，百分之百都能变成附属性工作。

随意归因呢，就是不把问题的根源追溯到业务上，寻找到技术上要解决的问题就停止了。我可以理解技术人员对业务领域的诸多陌生与百般不适，然后就在自己擅长的地方停止了。然而事实是，当我们在做业务系统的时候，最终是由业务为我们的软件买单的。那么如果既不能增效，又不能降本，那要我们有何用？难道是因为好看吗？

最后是迷信复用，它还有另一个和银弹很般配的名字，叫**金锤**。意思是说，我拿了锤子，然后满眼都是钉子，不问是啥就乱砸一气。实际上，这是在打着技术卓越（毕竟在讲复用了）的名义，不定义问题和随意归因。

Fred Brooks 论文中之所以用到银弹，是因为只有银弹可以杀死狼人。而狼人呢，是从一般人突然变身为恐怖的怪物。

正如软件项目，看着一切正常，突然就超期了，失控了，事故了。所以盲目追求将附属性工作的比例提高，而忽略本质性工作，这不是银弹，这是狼人的血。结果就是没有杀死狼人，反而自己变成了狼人。

好了，关于银弹就言尽于此吧。在进入下一个题外话之前，希望你能认真思考如下问题。

思考题

我们在旧约部分讲的内容里，既有理念，也有技巧。那么请按你的理解来分析一下，其中哪些帮我们更好地完成了本质性工作，哪些解决了附属性工作呢？

请把你的思考和想法分享在留言区，我会和你交流讨论。同时，也非常期待你能把自己的疑问和想听的话题，写在留言区，或者反馈给专栏编辑。我们下个题外话再见！

分享给需要的人，Ta订阅后你可得 20 元现金奖励

赞 2 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 说点题外话02｜模式并不是解决方案

下一篇 说点题外话04｜面向对象的原则适用于RESTful API吗？

精选留言 (10)

写留言

爱睡觉 置顶

2021-07-25

很有意思的是，“没有银弹”成为了一种银弹。
那就是面临任何一个不熟悉、不适应的方法，都可以高瞻远瞩的说一句“没有银弹，这个xxx不是万能的”。然后心安理得的继续原本的做法。
没有银弹就成了杀死一切改进建议的银弹

展开

作者回复: 没有银弹给的建议非常具体 解决本质性困难 除此之外别无他法

2

码农戏码 置顶
2021-07-26

可怜之人必有可恨之处

我们当前就是100%的投入附属性工作，还在加量加码，甚至整个行业风气都是，别人吹的牛，我们得实现了

...

展开

作者回复: to b复杂性反而在逻辑 to c在容量

HoshinoKanade
2021-07-24

能多說一下漸進增強嗎

展开

2

Oops!
2021-07-24

again, hard to sleep at night. 本课用了很多笔墨讲述两关联一循环的知识消化过程就是用来指导我们更好完成本质工作的。其他各类模式、分层和restful api 等都是解决附属工作中的一些技术性难点。

作者回复: 会心不远

2

狩月
2021-07-25

所以，低代码是行业毒瘤的原因也是不定义问题，随意归因，和迷信复用吧

作者回复: 逻辑上说 低代码过分强调附加性工作 回避回答是如何简化问题定义的

1 1

呵 哦 呵
2021-07-24

一针见血！

展开

1

马若飞
2021-07-27

本质性困难源于业务复杂性。如果业务极其简单，是否可认为银弹存在？

作者回复: 知识的传递也是本质性工作 所以也没有那么简单

云师兄
2021-07-25

再来一波题外话吧。

展开

编辑回复: 周二见

下弦の月
2021-07-24

补充适用于程序员的财务知识，有什么推荐的书籍或者学习套路么？

作者回复: 随便哪本都可以 貌似没有专门给程序员写的

切糕
2021-07-24

搞统一语言和领域建模就是解决本质性问题 具体落地就是后者啦